ソフトウェア検証における見える化とは

～第3者による検証活動の事例紹介～

2015/6/10

宇宙航空研究開発機構 (JAXA)
第三研究ユニット
梅田　浩貴

宇宙システムの種類とソフトウェア

人工衛星

- 姿勢制御系、データ処理系、ミッション系、スタートラッカー／GPSRなどセンサー系など組込み系が中心
- 最近は書換可能のものもある
- アセンブラ、C言語
- 再利用／シリーズ化
- 放射線の影響

ロケット

- 航法誘導制御系等
- 高信頼性／多数決
- アセンブラ、C言語
- ハードリアルタイム

宇宙ステーション

- 管制システム、マニュピュレータ、実験装置など膨大な数のソフトウエア
- 高い安全要求（2故障でも安全に！）
- コマンド、データ、処理の独立性
- システム構成にクルーが入る
- Ada、C言語

地上管制システム

- 管制システム等
- 安全性
- C、JAVA言語

目次

- ソフトウェアの検証とは
- アシュアランスケースの応用
- 「検証戦略」の可視化

問い

下記の者が同じ目的のソフトウェアを作成した場合品質は同じになると思いますか？

- A：実験等で十分なプログラミング経験を有した学生
- B：業務目的でソフトウェア設計経験がある企業の技術者

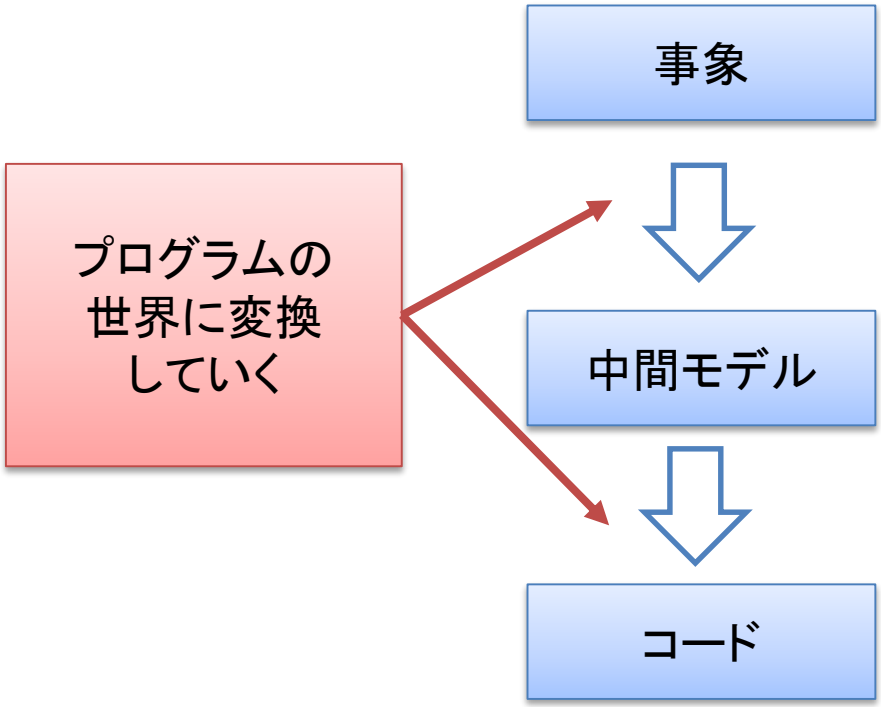
その理由は、何が想定されますか？

ソフトウェア検証とは

・(一般用語)「検証」とは、実際に物事に当たって調べ、仮説などを証明すること。

・ソフトウェア開発における「検証」とは、
ソフトウェア開発プロセスの各工程で作られるモデル(概念、数理、論理)が
それぞれ正しく変換されているかを確かめる作業

問題意識:　正しく変換されていることをどうやって確認したのか



ソフトウェアで実現したい事象
(ビジネス業務や機械の挙動など)

ユーザー要求から業務モデルなどの
中間モデルを生成し、
最終的に論理や計算などからなる
抽象モデル(コード)に置き換える作業

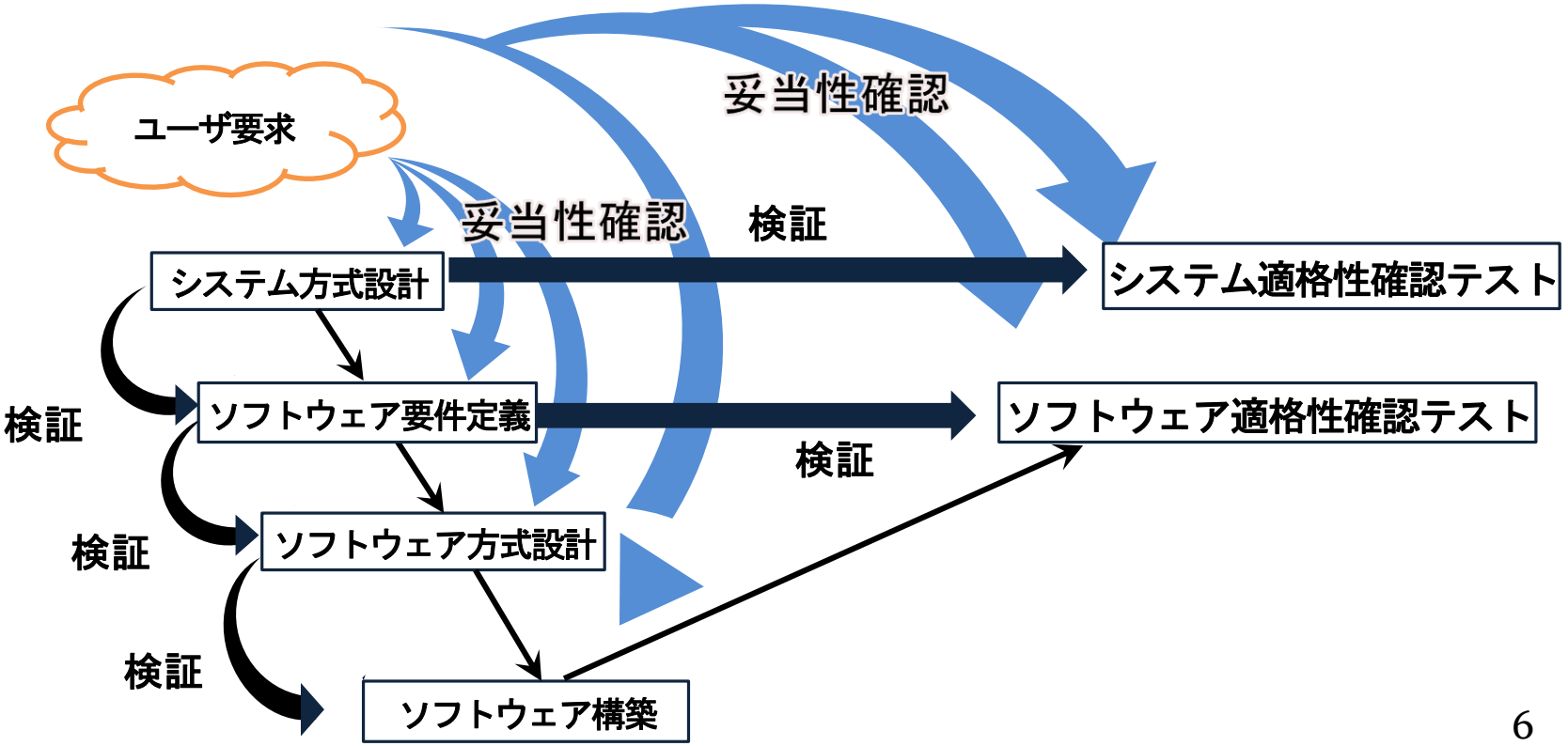
開発モデル(V字モデル)と検証

■妥当性確認と検証

・検証とは、前工程の仕様に基づいて確認すること

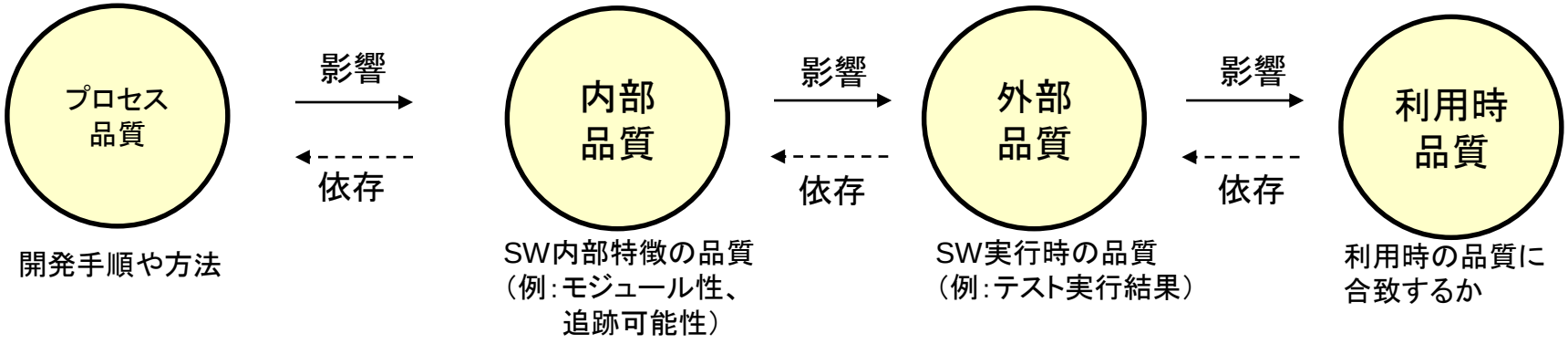
・妥当性とは、最終利用者の意図通りに動いているかどうかを確認すること

問題意識：　検証と妥当性を使い分けて効果的なレビューやテストはできているのか

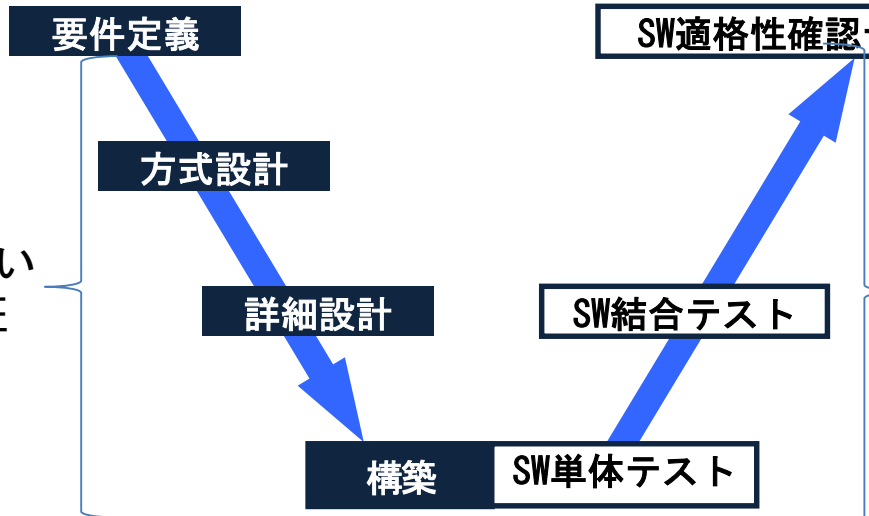


ソフトウェア品質とテスト

ソフトウェア品質モデル（ISO9126より）



欠陥が存在しない
（ソフトウェアに仕様にない
振舞がないこと）を保証



ソフトウエアテストは、
プログラムの実行によって
・正しく動作するか
・目標とした品質に到達しているか
・意図しない動作をしないかどうか
等を確認する作業
→ 欠陥の検出はできるが
欠陥がないことは証明不可

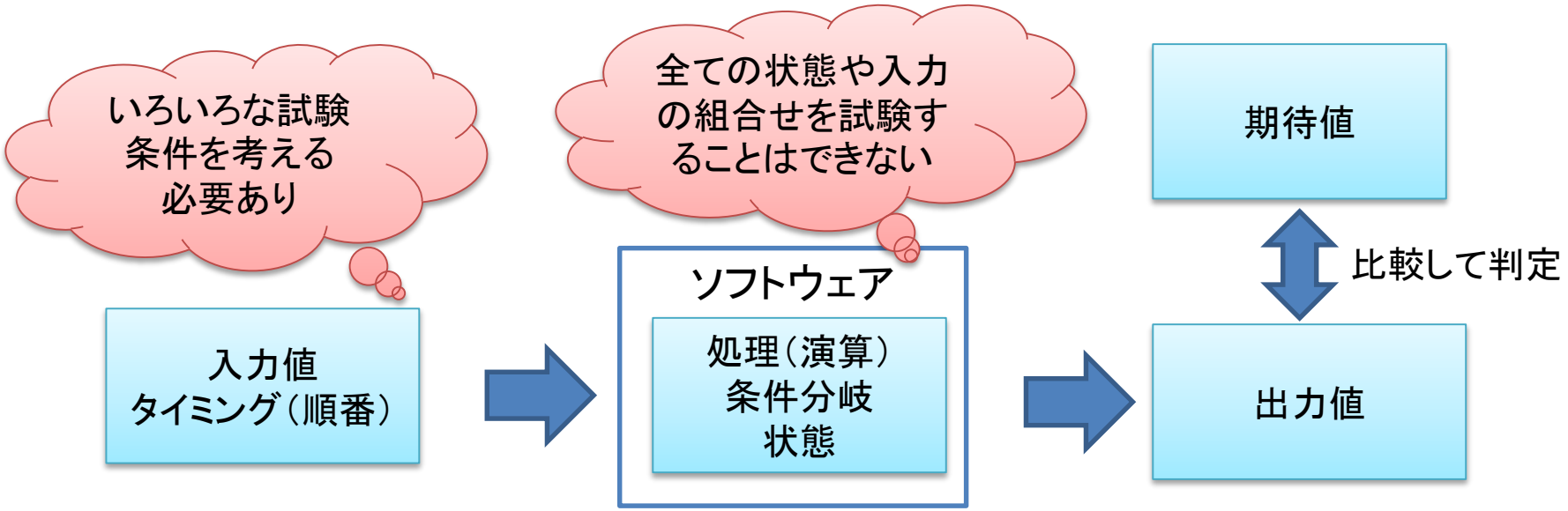
問題意識： ソフトウェアを高品質にする
＝ テストの網羅度上げるになっていないか。

ソフトウェアテストの概念と課題

問題意識：　ソフトウェアテストケースが多い
　　＝　網羅度が高い、有効なテストと判断していないか

・特徴1：ハードウェアは摩耗による故障があるため、
　　ソフトウェアは同じ条件によるテストを繰り返したところで効果がない。

・特徴2：　数学的に入力値と条件を網羅的にテストすることは、現実的に不可能



ソフトウェアテストの概念図

アシュアランスケースとは

アシュアランスケースとは

■アシュアランスケースとは
システムが保証すること（想定された状況下で動作する）を
議論によって構造化されたドキュメントで表現する方法

※アシュアランス（保証）をケース（論拠）する。

基本効果

可視化する

議論する

合意する

■品質とは

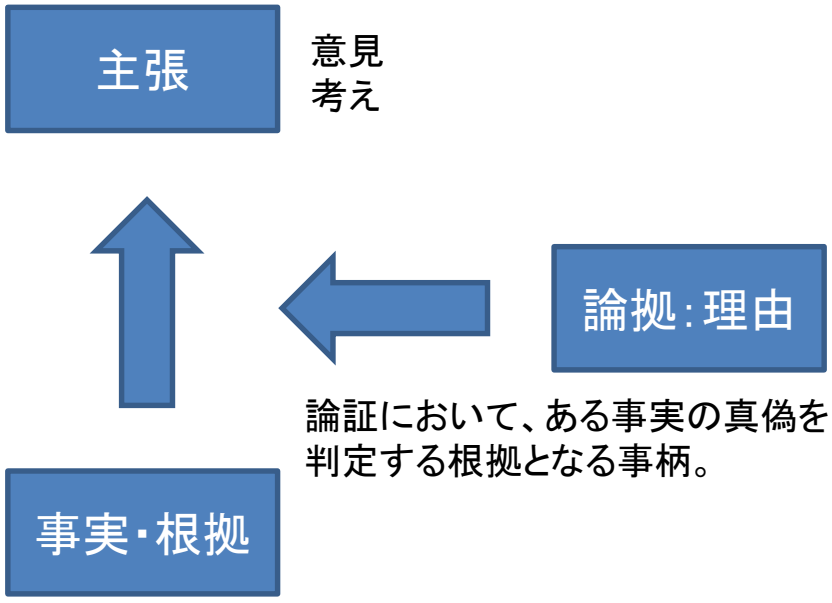
「品質とは誰かにとっての価値である」
（*Quality Software Management:*ジェラルド・ワインバーグ著）

可視化することで

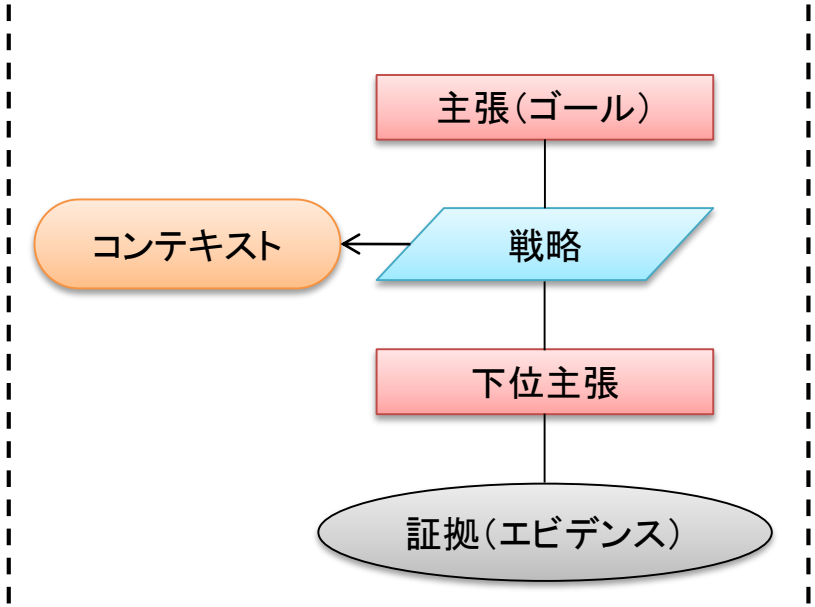
品質要求元と合意することが可能

トゥールミンロジックとGSN

トゥールミンロジック　＝　論証の組み立て方



GSN(Goal Structuring Notation)
＝　ゴール構造表記法



■GSNの効果

・階層化された複雑な議論を可視化し
　主張の曖昧さや抜け漏れを防ぐ。

・第三者への説明しやすさも高める。

ソフトウェア検証

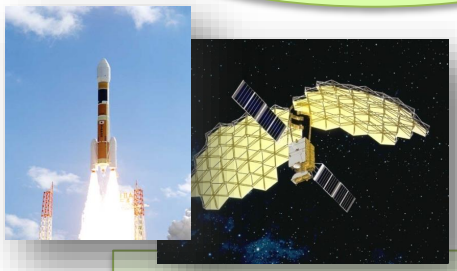
戦略の可視化

ソフトウェアテスト版のアシュアランスケースとは

ソフトウェアテストが、次工程へ進む上で十分に実施されていることを説明（保証）するために、「アシュアランスケース」を活用している。

説明先　　主張　　説明元

プロジェクトリーダー



保証対象：
ソフトウェア製品
＋成果物

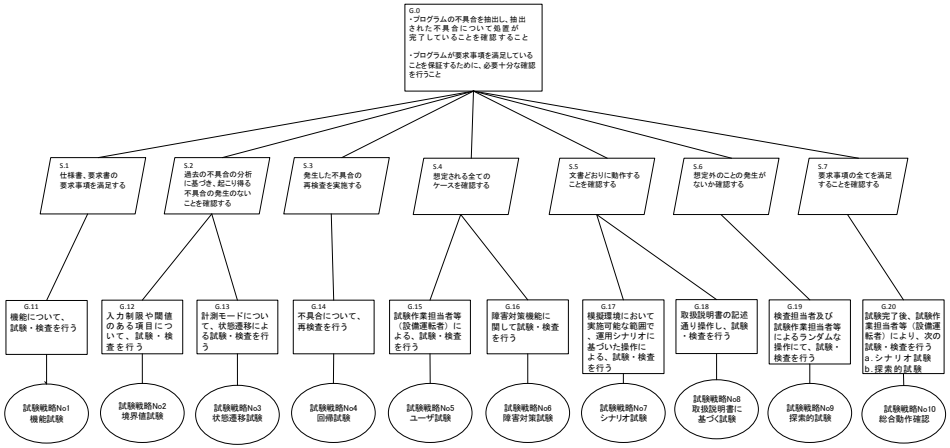
ソフトウェアテストが十分であること

テスト担当

証拠：エビデンス
（テスト仕様と実行結果）

ソフトウェアテスト版のアシュアランスケース

最初に作成したケース図の全体



G1:
プログラムの不具合を抽出し、抽出された不具合について処置が完了していることを確認すること

プログラムが要求事項を満足していることを保証するために、必要十分な確認を行うこと

※2階層まで抜粋

S.1
仕様書、要求書の
要求事項を満足する

S.2
過去の不具合の分析に基づき、起こり得る不具合の発生のないことを確認する

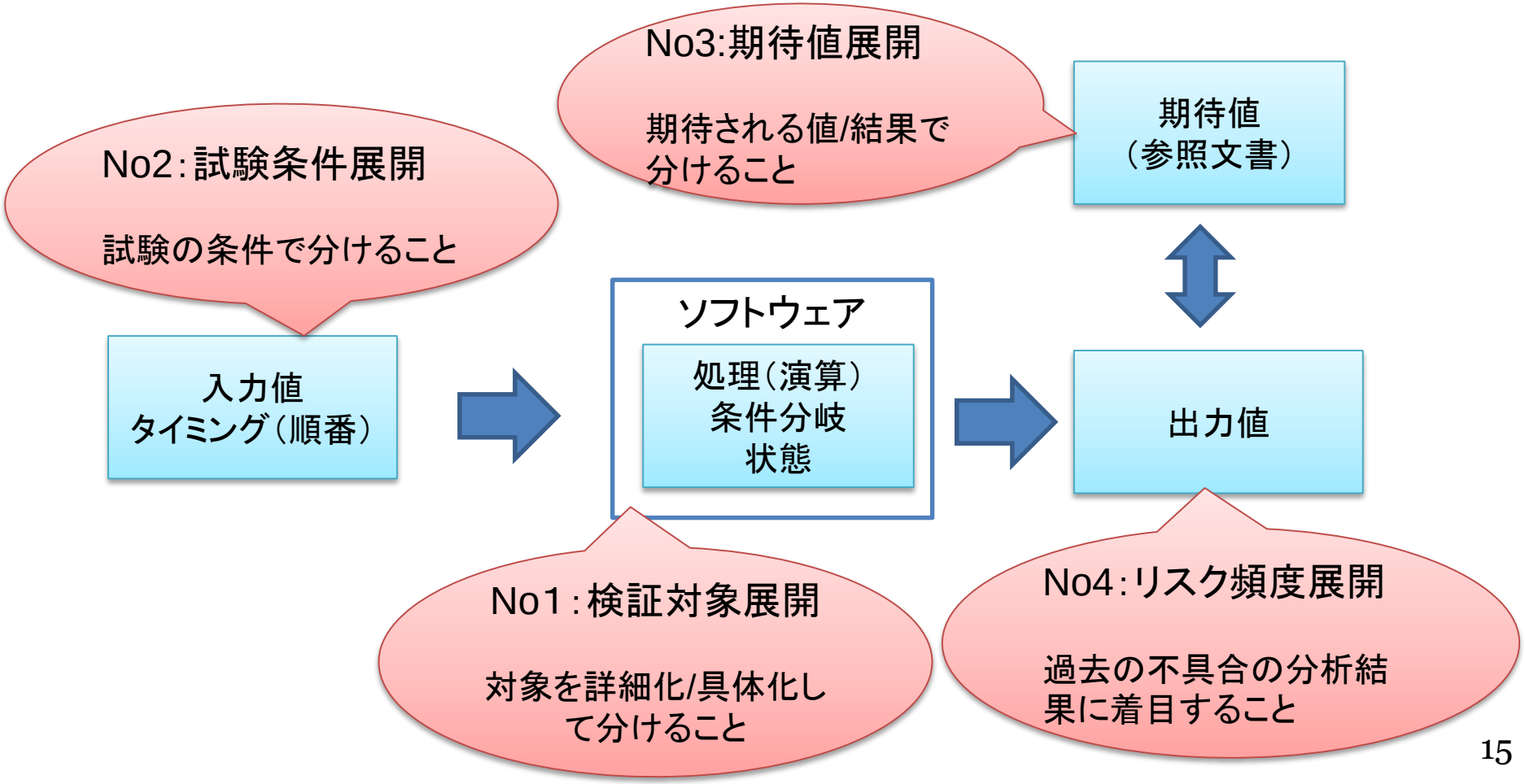
・・・・・

前提（コンテキスト）を使いこなせず、チェックリストのような展開

→ 論理的な議論の余地がない（＝GSNの効果がない）

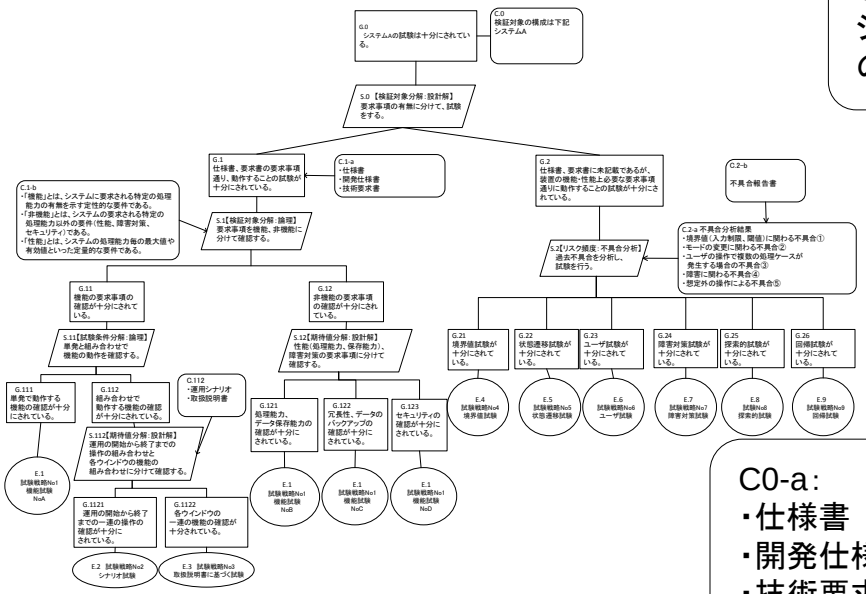
ソフトウェアテストにおける展開パターン

テストの効果を高めるため　論理モデルによる展開パターンを用意



ソフトウェアテスト版のアシュアランスケース

改善されたケース図の全体



※2階層だった構造が4階層に

何に基づいて試験をするのか

どこまで試験しているのか

議論が可能になった

C0-a:
システムAのソフトウェアの最終試験

G0: Aシステムは、試験が十分に実施されている。

S0:【検証対象分解:設計解】
要求事項の有無に分けて、試験をする。

C0-a:
・仕様書
・開発仕様書
・技術要求書

G1:仕様書、要求書の要求事項通り、動作することの試験が十分にされている。

・・・・

S1:【検証対象分解:論理】
要求事項を機能、非機能に分けて確認する。

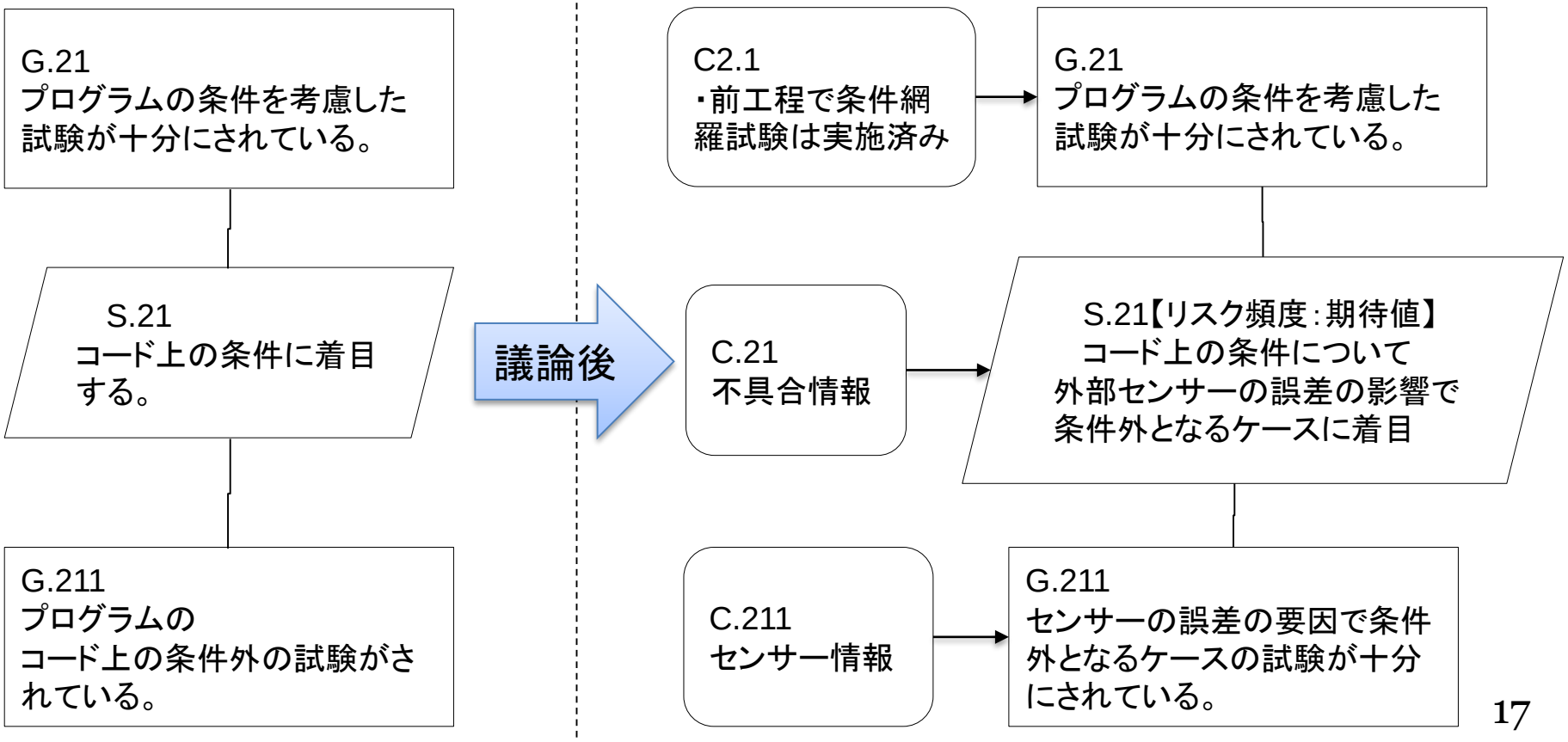
G1.1:機能の要求事項の確認が十分にされている。

G1.2:非機能の要求事項の確認が十分にされている。

ソフトウェアテスト版のアシュアランスケース

議論による効果（テスト精度向上）の例

議論により入力特性を定め、根拠に基づく試験ケースの設定へ



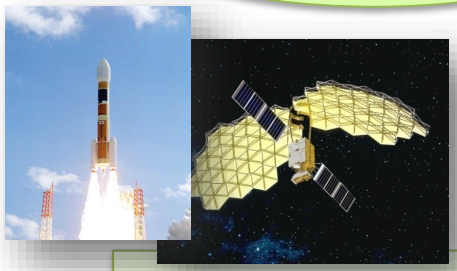
第3者による検証活動版のアシュアランスケースとは

第3者による検証サービスは、ソフトウェアに「リスク（不具合要因）」がないことを説明（保証）するために、「アシュアランスケース」を活用している。

※開発担当と異なる視点を持つことが必須のため。

説明先　　主張　　説明元

開発プロジェクト



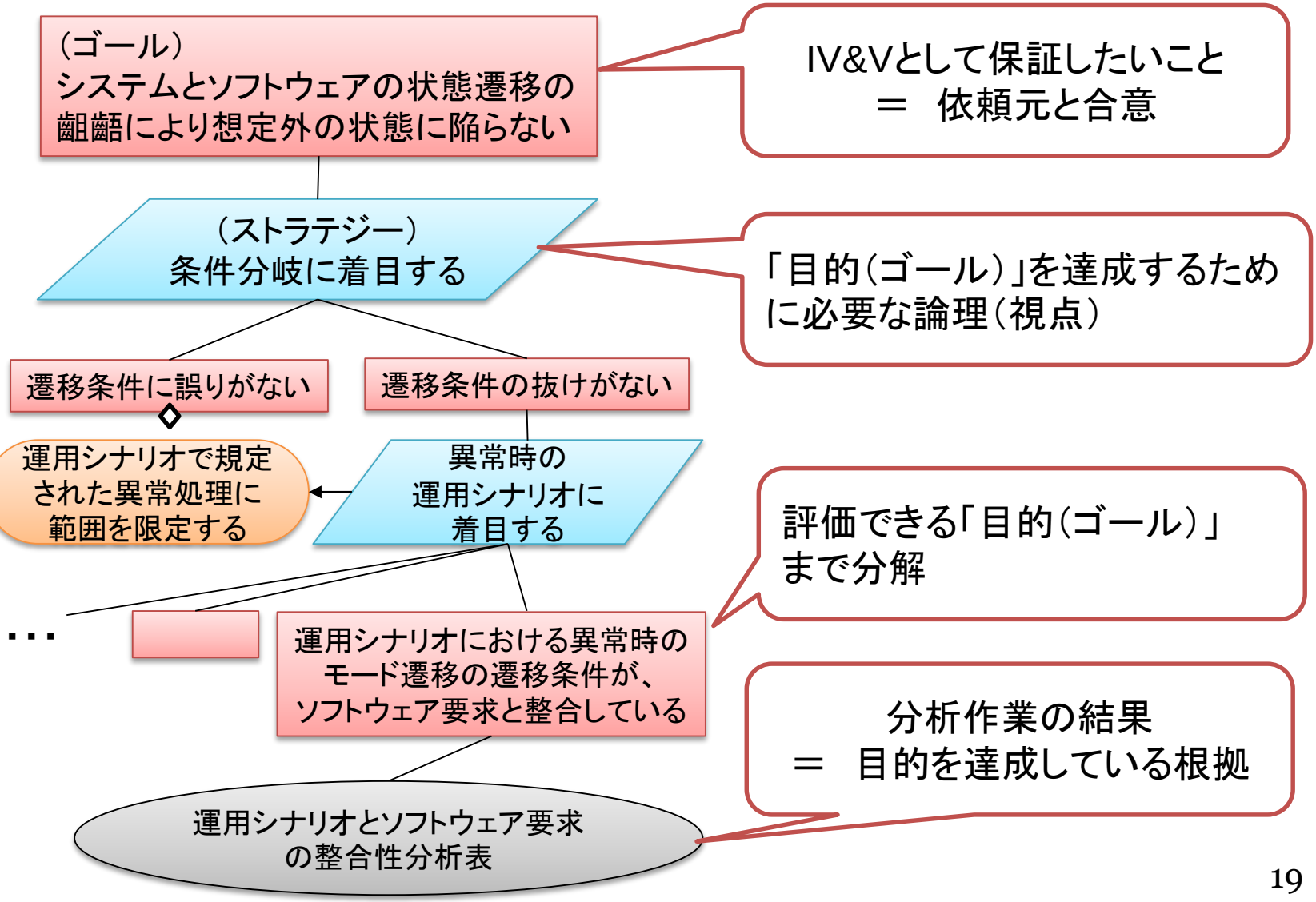
保証対象：
ソフトウェア製品
＋成果物

ソフトウェアリスク
がないこと

IV&V実施組織

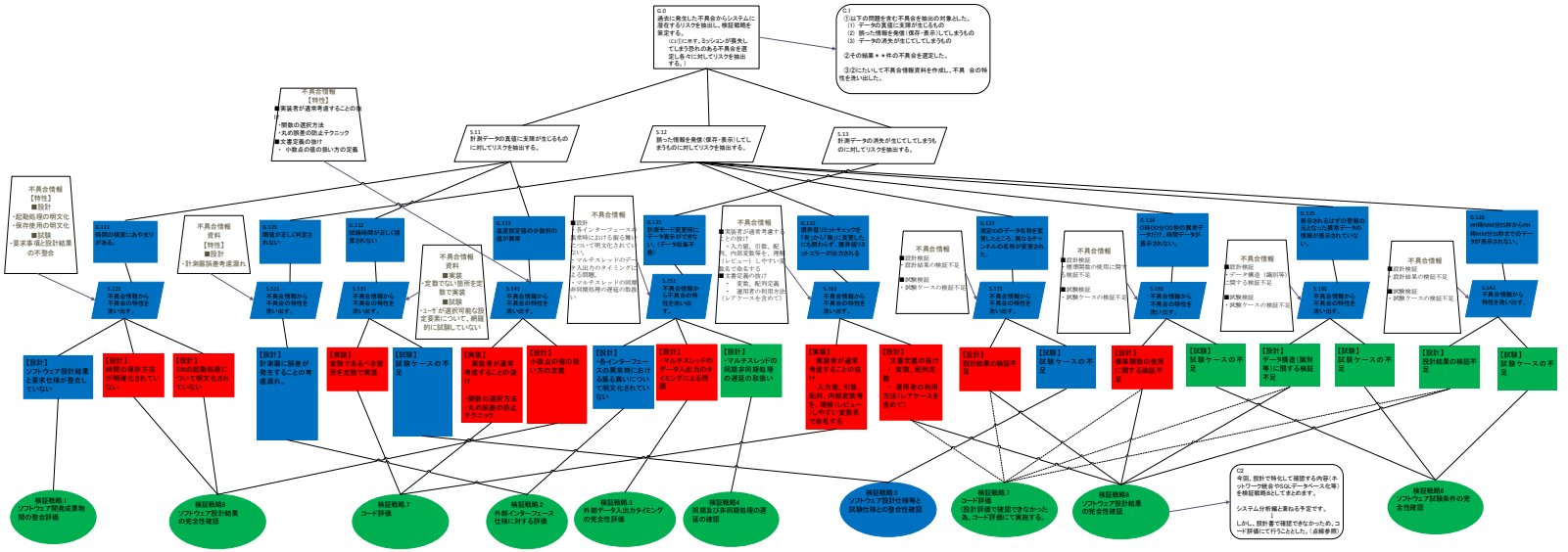
証拠：エビデンス
（IV&V活動の分析結果）

第3者による検証活動版のアシュアランスケース



ソフトウェアテストと第3者検証を合わせた確認事例

分類	例	図上の色
テスト担当が確認した内容	要求仕様にある各機能の動作 不具合水平展開による試験	テスト担当
第3者検証担当のみ確認した内容	設計検証（マルチタスクにおける競合防止の有無） コードレビュー（一貫した型定義や丸め誤差の影響有無）	第3者検証担当
テスト担当及び第3者検証ともに共通の確認内容	異常な外部入力値の試験	共通



まとめ

ソフトウェアテスト、第3者による検証と妥当性確認に対しアシュアランスケースを応用し、可視化や議論の事例をご紹介した。

ソフトウェア第3者検証活動
（IV&V：Independent Verification and Validation）
については、「IV&Vガイドブック」をご覧ください。

お問い合わせ先：IVV_INFO@jaxa.jp